Compte Rendu du TP1

Salaheddine ELMOHADEB

December 12, 2

Table des matières

1 Objectif

L'objectif de cette session pratique était de concevoir et d'implémenter un système de gestion des employés en Java, basé sur une architecture à plusieurs couches, incluant des modèles, des vues, des contrôleurs et des interactions avec une base de données.

2 Résumé des Travaux

Le travail assigné comprenait la création des composants suivants :

- **Modèle (Model)** : Une représentation des données des employés comprenant des attributs tels que le nom, l'email, le téléphone, le salaire, le poste et le rôle.
- **DAO (Data Access Object)** : Une couche permettant d'interagir avec la base de données PostgreSQL à l'aide de requêtes SQL.
- **Contrôleur (Controller)** : Une couche gérant la logique pour ajouter, supprimer, mettre à jour et afficher les employés.
- **Vue (View)** : Une interface graphique (GUI) permettant les interactions utilisateur.

3 Implémentation du Code

3.1 Classe Principale

```
package main;

    import DAO.EmployeImpl;
    import Model.EmployeModel;
    import View.EmployeView;
    import controller.EmployeController;

    public class main {

        public static void main(String[] args) {
                EmployeView View = new EmployeView();
                EmployeImpl DAO =new EmployeImpl();
                EmployeModel Model = new EmployeModel(DAO);
                new EmployeController(Model,View);
```

```
                View.setVisible(true);
        }
    }
```

3.2 DAO

3.2.1 Interface DAO

```
package DAO;
import java.util.List;
import Model.Employe;
import Model.Employe.Poste;
import Model.Employe.Role;

public interface EmployeI {

    Employe findById(int employeId);
    List<Employe> findAll();
    void add(Employe E);
    void update(Employe E,int id);
    void delete(int id);
    List<Role>findAllRoles();
    List<Poste>findAllPostes();


}
```

3.2.2 Implémentation DAO

```
package DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
```

```
import java.util.Arrays;
import java.util.List;

import Model.Employe.Role;

import Model.Employe;
import Model.Employe.Poste;


public class EmployeImpl implements EmployeI {
    private Connection conn;

    public EmployeImpl() {
        this.conn = connexion.getConnexion();
    }

    @Override
    public void add(Employe E) {
        String Query = "INSERT INTO Employee(nom, prenom, email,
--------telephone, salaire, role, poste)VALUES(?,?,?,?,?,?,?)";

        try (PreparedStatement stmt = conn.prepareStatement(Query)) {
            stmt.setString(1, E.getNom());
            stmt.setString(2, E.getPrenom());
            stmt.setString(3, E.getEmail());
            stmt.setString(4, E.getTelephone());
            stmt.setDouble(5, E.getSalaire());
            stmt.setString(6, E.getRole().name());
            stmt.setString(7, E.getPoste().name());
            stmt.executeUpdate();
            System.out.println("Employe ajoute avec succes!");
        } catch (SQLException e) {
            System.out.println("Erreur lors de lajout de lemploy !")
            //e.printStackTrace();

        }
    }
```

4 Résultats

L'application a été testée sous Ubuntu avec PostgreSQL comme système de gestion de base de données. L'implémentation a permis de démontrer les fonctionnalités suivantes :

- Établir une connexion à une base de données PostgreSQL.
- Ajouter, supprimer et mettre à jour les enregistrements des employés via des interactions avec l'interface graphique.
- Afficher les données des employés dans un tableau au sein de l'interface graphique.

5 Défis et Solutions

- **Défi :** Gestion des exceptions SQL lors des opérations sur la base de données.
  Solution : Ajout de blocs try-catch et de messages d'erreur appropriés.
- **Défi :** Synchronisation des mises à jour de l'interface avec les modifications dans la base de données.
  Solution : Implémentation d'une méthode pour actualiser dynamiquement le tableau après chaque opération.

6 Conclusion

Cette session a permis d'acquérir une expérience pratique dans la création d'une application Java complète avec une interface graphique et une intégration à une base de données. Elle a renforcé la compréhension de l'architecture MVC et des interactions avec une base de données SQL.